In the Claims:

Please amend claims 1, 4, 6, 7, 11, 14, 17, 21, 26, 27 and 29. The claims are as follows:

1. (Original) A circuit, comprising:

multiple ~~W-bit~~ packet data slice latches each having W-bits where W is a positive integer, each packet data slice latch having inputs and outputs, said packet data slice latches connected in series from a first to a last packet data slice latch, outputs of a previous packet data slice latch connected to inputs of an immediately subsequent packet data slice latch;

a data partition comprising multiple data XOR subtree levels and having data latches between said data XOR subtree levels, said data partition having inputs and outputs, said outputs of each packet data slice latch connected to corresponding inputs of said data partition;

a remainder partition comprising multiple remainder XOR subtree levels and having remainder latches between said remainder XOR subtree levels, said remainder partition having inputs and outputs;

a combinatorial XOR tree having inputs and outputs, the outputs of said remainder partition and the outputs of said data partition connected to corresponding inputs of said combinatorial XOR tree; and

~~an M-bit~~ a current cyclic redundancy check (CRC) remainder latch having M-bits where M is a positive integer and having inputs and outputs, the outputs of said combinatorial XOR tree connected to corresponding inputs of said current CRC remainder latch and the outputs of said current CRC remainder latch connected to corresponding inputs of said remainder partition.

2. (Original) The circuit of claim 1, wherein:

each output of each of said packet data slice latches is connected to a corresponding input of a lowest XOR subtree level of said data partition and each intermediate XOR subtree level of said data partition is connected between an immediately higher XOR subtree level of said data partition through an intervening latch level and an immediately lower XOR subtree level of said data partition through an intervening latch level; and

each output of said CRC remainder latch is connected to a corresponding input of a lowest XOR subtree level of said remainder partition and each intermediate XOR subtree level of said remainder partition is connected between an immediately higher XOR subtree level of said remainder partition through an intervening latch level and an immediately lower XOR subtree level of said remainder partition through an intervening latch level.

3. (Original) The circuit of claim 1, wherein said packet data slice latches, all said data latches in said data partition, all said remainder latches in said remainder partition and said current CRC remainder latch are responsive to the same clock signal.

4. (Currently Amended) The circuit of claim 1, wherein B is <u>a positive integer</u> and ~~the~~ is the maximum number of XOR operations to be performed in a single remainder XOR subtree level of said remainder partition and A[[,]]<u>is a</u> ~~the~~ number of remainder XOR subtree levels of said remainder partition <u>and A</u> is <u>also</u> the smallest ~~whole~~ positive integer greater than the log to the base B of the largest number of bits I of a subset of the M-bits of said CRC remainder latch.

5. (Original) The circuit of claim 1, wherein each packet data slice latches sequential and different data slices.

6. (Currently Amended) The circuit of claim 5, wherein the maximum number of inputs to any data XOR subtree in any data XOR subtree level of said data partition is B, where B is a positive integer.

7. (Currently Amended) The circuit of claim 6, wherein the number of levels of said data partition [[(Y-1),]] is the smallest ~~whole~~ positive number greater than the log to the base B of B times J, where J is the largest number of bits ~~J~~ of a subset of the W-bits of said packet data slice latch.

8. (Original) The circuit of claim 7, wherein W=2048, I=20, J=1059, A=3 and B=3.

9. (Original) The circuit of claim 1, wherein:

a number of data XOR subtree levels in said data partition is equal to a number of remainder XOR subtree levels in said remainder partition.

10. (Original) The circuit of claim 1, wherein the slowest data XOR subtree level of said data partition is no slower than the slowest remainder XOR subtree level of said remainder partition.

11. (Currently Amended) A method for performing a cyclic redundancy check, comprising:

providing multiple ~~W-bit~~ packet data slice latches each having W-bits where W is a positive integer and each packet data slice latch having inputs and outputs[[,]]

connecting said packet data slice latches ~~connected~~ in series from a first to a last packet data slice latch, outputs of a previous packet data slice latch connected to inputs of an immediately subsequent packet data slice latch;

providing a data partition comprising multiple data XOR subtree levels and having data latches between said data XOR subtree levels, said data partition having inputs and outputs[[,]];

connecting said outputs of each packet data slice latch ~~connected~~ to corresponding inputs of said data partition;

providing a remainder partition comprising multiple remainder XOR subtree levels and having remainder latches between said remainder XOR subtree levels, said remainder partition having inputs and outputs;

providing a combinatorial XOR tree having inputs and outputs[[,]];

connecting said ~~the~~ outputs of said remainder partition and the outputs of said data partition ~~connected~~ to the inputs of said combinatorial XOR tree; ~~and~~

providing ~~an M-bit~~ a current cyclic redundancy check (CRC) remainder latch having M-bits where M is a positive integer and having inputs and outputs[[,]];

connecting the output of said combinatorial XOR tree ~~connected~~ to the inputs of said current CRC remainder latch and the outputs of said current CRC remainder latch to the inputs of said remainder partition; and

presenting a data packet to said inputs of said packet data slice latches and outputting a CRC remainder at said outputs of said CRC remainder latch.

12. (Original) The method of claim 11, wherein:

each output of each of said packet data slice latches is connected to a corresponding input of a lowest XOR subtree level of said data partition and each intermediate XOR subtree level of said data partition is connected between an immediately higher XOR subtree level of said data partition through an intervening latch level and an immediately lower XOR subtree level of said data partition through an intervening latch level; and

each output of said CRC remainder latch is connected to a corresponding input of a lowest XOR subtree level of said remainder partition and each intermediate XOR subtree level of said remainder partition is connected between an immediately higher XOR subtree level of said remainder partition through an intervening latch level and an immediately lower XOR subtree level of said remainder partition through an intervening latch level.

13. (Original) The method of claim 11, wherein said packet data slice latches, all said data latches in said data partition, all said remainder latches in said remainder partition and said current CRC remainder latch are responsive to the same clock signal.

14. (Currently Amended) The method of claim 11, wherein B is <u>a positive integer</u> and ~~the~~ is the maximum number of XOR operations to be performed in a single remainder XOR subtree level of said remainder partition and A[[,]]<u>is a</u> ~~the~~ number of remainder XOR subtree levels of said remainder partition <u>and A</u> is <u>also</u> the smallest ~~whole~~ positive integer greater than the log to the base B of the largest number of bits I of a subset of the M-bits of said CRC remainder latch.

15. (Original) The method of claim 11, wherein each packet data slice latches sequential and different data slices.

16. (Currently Amended) The method of claim 15, wherein the maximum number of inputs to any data XOR subtree in any data XOR subtree level of said data partition is B, where B is a positive integer.

17. (Currently Amended) The method of claim 16, wherein the number of levels of said data partition [[(Y-1),]] is the smallest ~~whole~~ positive number greater than the log to the base B of B times J, where J is the largest number of bits ~~J~~ of a subset of the W-bits of said packet data slice latch.

18. (Original) The method of claim 17, wherein W=2048, I=20, J=1059, A=3 and B=3.

19. (Original) The method of claim 11, wherein a number of XOR operations in said data partition is equal to a number of XOR operations in said remainder partition.

20. (Original) The method of claim 11, wherein the slowest data XOR subtree level of said data partition is no slower than the slowest remainder XOR subtree level of said remainder partition.

21. (Currently Amended) A method of designing a circuit, the method comprising:

(a) providing a cyclic redundancy check (CRC) circuit design for a current CRC remainder, comprising:

outputs of a packet data slice latch connected to inputs of a data XOR tree;

outputs of a current CRC remainder latch connected to inputs of a remainder XOR tree; and

outputs of said data XOR tree and outputs of said remainder XOR tree coupled to corresponding inputs of said current CRC remainder latch through a combinatorial XOR tree;

(b) substituting a previous CRC cycle data and corresponding previous CRC remainder for said current CRC remainder or for a previously subsituted CRC remainder and adding an additional packet data slice latch, an additional CRC remainder latch, an additional data XOR tree, an additional remainder XOR tree and an additional combinatorial XOR tree to said CRC circuit design without altering the a CRC remainder result of said CRC circuit design;

(c) partitioning all packet data slice latches and all data XOR trees into a data partition and all additional current CRC remainder latches and all remainder XOR trees into a remainder partition;

(d) combining all remainder XOR trees into a single remainder XOR tree and combining all data XOR trees into a single data XOR tree;

(e) repeating steps (b) through (c) a predetermined number of times; and

(f) distributing said single remainder XOR tree in said remainder partition over two or more remainder XOR subtree levels, distributing all additional CRC remainder latches over one or more remainder latch levels, distributing said single data XOR trees in said data partition over two or more data XOR subtree levels.

22. (Original) The method of claim 21, wherein step (f) further includes placing remainder latch levels between each remainder XOR subtree level of said remainder partition and a placing a data latch level between each data XOR subtree level of said data partition.

23. (Original) The method of claim 21, wherein a number of XOR operations in said data partition is equal to a number of XOR operations in said remainder partition.

24. (Original) The method of claim 21, wherein step (f) further includes selecting each data XOR subtree to be no slower than the slowest remainder XOR subtree.

25. (Original) The method of claim 21, further including:

(g) connecting a clock input of each data latch in said data partition, each remainder latch in said remainder partition, said current CRC remainder latch and said packet data slice latch to a same clock input pin.

26. (Currently Amended) A method of designing a circuit, the method comprising:

(a) distributing a current cyclic redundancy check (CRC) remainder XOR calculation of ~~an M-bit~~ a redundancy check circuit into a remainder partition comprising multiple levels of remainder XOR subtrees and having remainder latches between said levels of remainder XOR subtrees; and

(b) distributing a packet data slice XOR function of said ~~M-bit~~ redundancy check circuit into a data partition of comprising multiple levels of data XOR subtrees and having data latches between said levels of data XOR subtrees.

27. (Currently Amended) The method of claim 26 wherein step (a) includes:

(a1) determining I, [[the]] a largest number of bits of a subset of ~~M bits~~ a number of bits of a current CRC remainder ~~of a CRC result~~ required to generate an output ~~bits of~~ on said remainder partition;

(a2) determining B, [[the]] a maximum number of XOR operations to be performed in a single remainder XOR subtree level of said remainder partition;

(a3) calculating A, [[the]] a number of XOR subtree levels in said remainder partition;

(a4) substituting (A-1) cycles of CRC operation into a current CRC remainder calculation;

(a5) distributing said current CRC remainder XOR calculation among remainder XOR subtrees of said remainder partition such that no remainder XOR subtree level has more than B inputs; and

(a6) inserting a remainder latch level between each remainder XOR subtree level.

28. (Original) The method of claim 27, wherein step (b) includes:

(b1) distributing said packet data slice XOR function among data XOR subtrees of said data partition such that no data XOR subtree level has more than B inputs; and

(b2) inserting a data latch level between each data XOR subtree level of said data partition.

29. (Currently Amended) The method of claim 28, wherein a number of levels of said packet data slice XOR partition [[(Y-1)]] is the smallest whole positive number greater than the log to

the base B of B times ~~J-bits where J-bits is the~~ , where J is the number of bits in a largest data slice ~~expected~~ said circuit is designed to handle.

30. (Original) The method of claim 26, further including:

(a7) calculating values of previous CRC remainders and corresponding previous packet data slices for j = (A-1) cycles that will result in a j = 0 cycle value that will be a used as an initial CRC remainder value.